BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Pada kerja magang kali ini, penulis berkesempatan untuk menjadi bagian dari departemen *Digital Workplace DevOps* dengan posisi sebagai *automation tester*. Tugas magang diberikan oleh *mentor* dan dalam pelaksanaannya, seluruh pekerjaan yang dilakukan dibimbing dan dievaluasi langsung oleh *mentor* yang merupakan senior *test engineer* di CIMB Niaga, Bapak Alvin Ega Variandi. *Mentor* terlebih dahulu memberikan latar belakang aplikasi yang akan di *testing* kemudian memberikan referensi pembelajaran mengenai *automation testing* menggunakan *framework* Cucumber BDD. Dalam pelaksanaan program magang ini juga diikuti oleh mahasiswa dari universitas lainnya yang tergabung dalam satu tim *development* khusus magang. Anggotanya terdiri dari seorang *tester* (penulis), seorang *scrum master* yang memimpin alur *scrum development* dan lima orang *developer*.

Berhubung himbauan *work from home* masih berlaku, selama program kerja magang berlangsung, bimbingan, arahan pantauan dari pengawas lapangan terus dilakukan untuk meningkatkan pemahaman mengenai *feature-feature* yang diuji pada aplikasi Octo Smart serta *sprint planning* khusus tim magang yang berjalan selama program magang berlangsung. Selama bekerja, *progress* harus selalu dilaporkan kepada *mentor* dan *scrum master* setiap hari pada *daily meeting*

melalui platform Google Meet, serta *weekly review* dengan supervisi melalui platform Cisco Webex, sehingga apabila terdapat kesalahan, kendala atau hal yang tidak sesuai dengan spesifikasi, dapat langsung diperbaiki atau didiskusikan. Begitu pula jika terdapat pertanyaan-pertanyaan ataupun terjadi kesulitan saat pengerjaan, dapat langsung bertanya kepada *mentor*.

3.2. Tugas yang Dilakukan

Berikut merupakan rincian tugas yang dilakukan selama pelaksanaan kerja magang dalam waktu 122 hari atau total 27 minggu di PT. Bank CIMB Niaga.

Tabel 3.1 Rincian Tugas Magang

Pekerjaan yang Dilakukan	**Rincian**	**Minggu Ke-**
Pengenalan Lingkungan Kerja	● Pengenalan awal perusahaan dan *working culture* perusahaan ● Pengenalan tim *intern* dan *job desc* masing-masing	1-2
Eksplorasi Penggunaan *Software* Katalon Studio	● Pengenalan Katalon Studio untuk *mobile testing* ● Mempelajari dan latihan menulis *test scenario* dan *test case* menggunakan *framework* Cucumber BDD dan bahasa Gherkin ● Belajar mengeksekusi *test case* pada aplikasi *dummy* berbasis *mobile* ● Pengenalan SDLC perusahaan	3-4

Pekerjaan yang Dilakukan	**Rincian**	**Minggu Ke-**
Pengenalan *Testing Environment* di Octo Smart	● Belajar dasar membuat *custom test rules* untuk menjalankan *test case* pada aplikasi *mobile* ● Melakukan eksplorasi repositori aplikasi Octo Smart ● Mempelajari *flow* dan *behavior* aplikasi serta *feature-feature* dasar Octo Smart	5-6
Aktifitas *Automation Testing* Terhadap *Feature* Octo Smart	● Memulai *assignment* untuk melakukan otomatisasi *feature* pada Octo Smart ● Membuat *test scenario* dan *test case* untuk *feature-feature* utama pada Octo Smart (total *feature* adalah 22 *feature*)	7-21
Melakukan *Adjustment Fix*	● Melakukan *adjustment fix* terhadap *feature* yang telah di otomatisasi terhadap aplikasi dengan versi yang baru	22-23
Belajar Melakukan API *Testing*, Memulai *Sprint* Baru	● Memulai *sprint planning* pada *project* untuk tim magang ● Mempelajari dasar melakukan API *Testing* menggunakan Postman ● Membantu melakukan API *Testing* pada *user story* yang telah ditentukan	24-26
Demo *Testing*	● Melakukan Demo *Project* Pada Tim Internal CIMB Bersama Tim Magang	27

3.2.1. Pengenalan Lingkungan Kerja (Minggu 1-2)

Proses pelaksanaan magang pada minggu awal dimulai dengan perkenalan pada perusahaan, divisi, dan *job description* yang akan dijalani selama 6 bulan kedepan. Perkenalan awal ini dipimpin oleh Pak Yonathan selaku supervisi atau kepala divisi *Digital Workplace DevOps* di CIMB Niaga. Kemudian, tim *intern* pun diresmikan yang terdiri dari *developer*, *scrum master*, dan *tester* yang masing-masing akan dibimbing oleh para mentor pada bagiannya.

Pengenalan lingkungan dan budaya perusahaan dilakukan oleh arahan HR dan supervisi divisi. Dikarenakan belum dapat bekerja secara langsung di kantor, pengenalan dilakukan melalui aplikasi LoG CIMB Niaga. Yaitu sebuah aplikasi yang disediakan perusahaan untuk karyawan CIMB yang disebut CIMBian dalam mempelajari lingkungan dan budaya perusahaan lebih dalam serta lengkap. Di dalamnya berisikan modul-modul serta evaluasi yang harus diselesaikan. Modul tersebut berisi seperti sejarah perusahaan, peraturan perusahaan, kode etik, dsb. Berikut adalah contoh tampilan beranda aplikasi LoG CIMB Niaga.

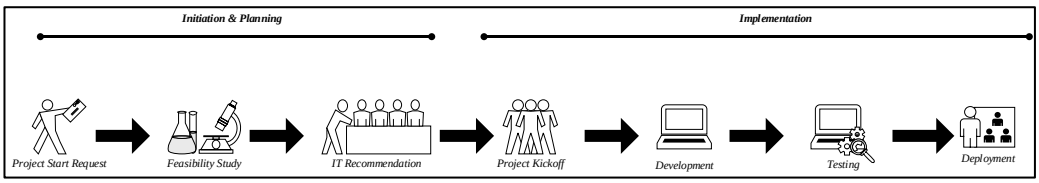
Gambar 3.1. Tampilan Beranda LoG CIMB Niaga

3.2.2. Eksplorasi Penggunaan *Software* Katalon Studio (Minggu 3-4)

Setelah melalui fase perkenalan, mentor memberikan tugas untuk mempelajari dasar-dasar penggunaan Katalon Studio untuk *automation testing* begitu juga dengan referensi-referensi pembelajarannya secara bertahap. Dari cara melakukan instalasi Katalon Studio, mengelola *workspace*, merekam objek dalam aplikasi *dummy* berbasis *mobile*, hingga menulis dan mengeksekusi *test case* sederhana. Setiap hasil latihan yang dikerjakan akan diberikan kepada mentor untuk diperiksa dan latihan ini

akan terus berlanjut hingga penulis menguasai betul dan terbiasa dengan environment Katalon Studio.

Di sisi lain, *scrum master* bertugas untuk memantau perkembangan *sprint* yang pada awalnya ditugaskan kepada tim *developer* pada proyek yang berbeda. Setiap hari, *scrum master* mengadakan sebuah *daily meeting* melalui Google Meet untuk berdiskusi pada semua tim *intern* mengenai *progress* pekerjaan serta kendala yang dihadapi. Ia juga bertugas untuk menyediakan *collaboration space* menggunakan Trello untuk memantau *progress* pekerjaan tim. Selain itu, setiap minggunya, penulis juga wajib untuk mengikuti *weekly meeting* khusus untuk tim *automation* di CIMB Niaga. Tujuan *weekly meeting* ini kurang lebih sama dengan *daily meeting* dimana setiap *squad* yang terdiri dari *test*er menginformasikan *progress* dan kendala pada pekerjaan mereka selama seminggu terakhir.



Gambar 3.2. *IT Process Development Lifecycle* Yang Ada Di Perusahaan

Gambar 3.1. menunjukkan sebuah PDLC yang ada di perusahaan. Di akhir minggu ke-4 para mentor mengadakan sebuah *sharing session* bersama tim *intern* untuk membahas SDLC yang ada di perusahaan (istilah perusahaan menggunakan PDLC).

Pada tahapan pertama terdapat sebuah PSR (*Project Start Request*) yang perlu dibuat oleh seorang *Project Manager* kemudian di approve oleh *Product Owner*. Selanjutnya dilakukan sebuah *feasibility study* oleh *Product Owner,* Developer, dan tim UI/UX. Pada tahapan tersebut membahas rancangan produk yang akan dibangun dari sisi *developer* dan *design*er. Kemudian tahapan selanjutnya adalah IT *Recommendation*. Tahapan ini dilakukan oleh *Product Owner* dan tim UI/UX untuk mengkonfirmasi desain yang telah mereka rancang dan apabila terdapat rekomendasi dari tim *Product* maka akan dilakukan revisi lagi.

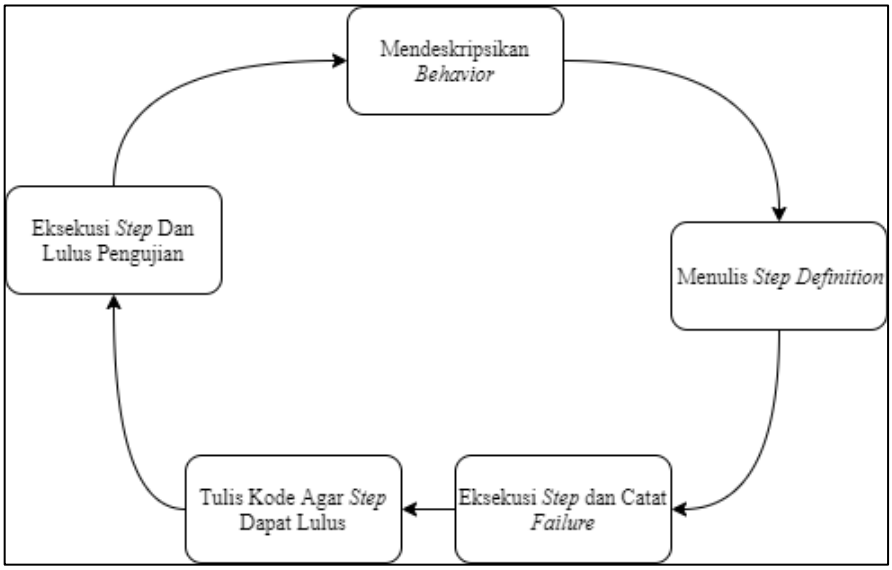
Apabila semua pihak (*dev*, *product*, dan UI/UX) sudah mencapai kesepakatan megenai produk yang akan dibangun, barulah dilakukan *Project Kickoff*. Tahapan ini dimulai dengan adanya Sprint Planning yang dipimpin oleh Scrum Master. Dari hasil *sprint* planning tim *developer*, UI/UX, dan *product* akan melakukan *daily sprint*, *grooming*, *review*, dan *sprint* planning kembali hingga semua *user story* terselesaikan. Setelah tahapan *Development* selesai, *test*er akan melakukan *Testing* terhadap produk berupa SIT dan UAT (pada akhir minggu). Tahapan terakhir adalah *Deployment* yang meliputi tim *product*, *developer*, dan *test*er. Kegiatan *Deployment* terbagi menjadi *Rehearsal*, *Smoke Test*, CCRB, dan *Promote*.

3.2.3. Pengenalan *Testing Environment* di Octo Smart (Minggu 5-6)

Setelah terbiasa dengan software Katalon Studio, mentor memberikan materi tambahan sebelum akhirnya memberikan aplikasi yang sesungguhnya untuk diuji. Materi tersebut berkaitan dengan penggunaan

sebuah *framework* bernama Cucumber BDD (*Behavior Driven Development*). *Framework Behavior Driven Development* (BDD) adalah proses pengembangan *software* yang merupakan cabang dari *framework Test* Driven *Development* (TDD) dan juga merupakan salah satu bagian dari *agile testing*. Singkatnya, dengan adanya *framework* ini dapat memberikan keunggulan bagi *tester* untuk membuat *test script* dari kedua perspektif *developer* dan juga *customer*.

Di awal projek, *developer*, *project manager*, *QA*, dan *UAT tester*, serta *product owner* bersama-sama melakukan *brainstorm* tentang scenario uji mana yang harus lulus agar aplikasi yang dikerjakan dapat dibilang sukses. Sementara itu, Cucumber adalah salah satu *open source tools*, yang mendukung *framework* BDD. Lebih tepatnya, Cucumber dapat didefinisikan sebagai sebuah *framework* untuk *testing* yang didorong oleh teks bahasa Inggris biasa dan dapat berfungsi sebagai dokumentasi, tes otomatis, dan bantuan pengembangan lainnya [3]. Berikut adalah *testing cycle* menggunakan *framework* BDD.

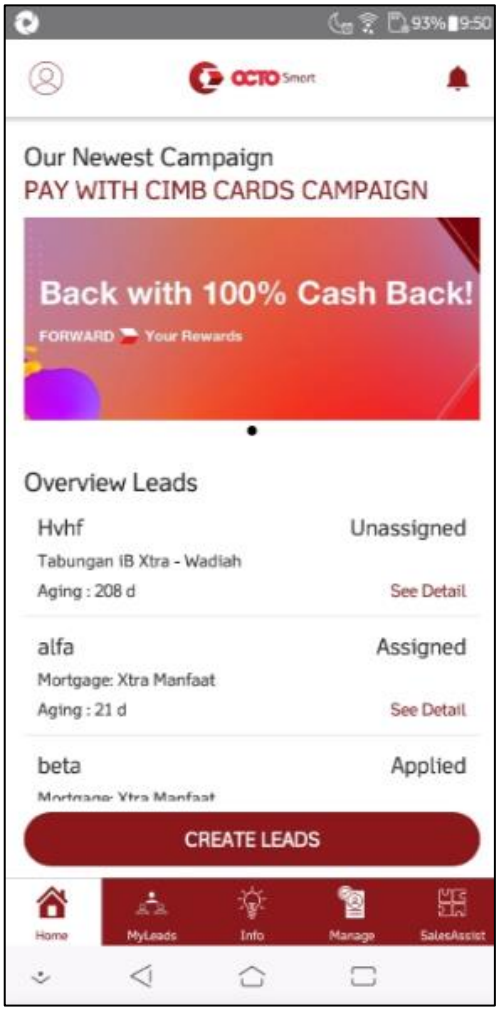


Gambar 3.3. *Test Cycle* Pada *Framework* BDD

Pada saat kode siap, *test script* juga sudah siap. Kode harus lulus *test script* yang ditentukan dalam BDD. Jika tidak terjadi, *refactoring* kode akan dibutuhkan. Kode akan dibekukan hanya setelah eksekusi *test script* yang ditentukan berhasil. Cucumber disini, bertugas untuk membaca kode yang ditulis dalam teks bahasa Inggris biasa (Bahasa Gherkin) di file *feature* [4].

Setelah menguasai dasar penulisan *test script* dan *scenario* menggunakan konsep BDD, mentor memberikan aplikasi perusahaan yang ditujukan untuk pengujian otomatisasi bernama Octo Smart. Octo Smart merupakan sebuah aplikasi berbasis Android untuk karyawan dan mitra kerja PT. CIMB Niaga Tbk untuk menjembatani interaksi antar karyawan dan meningkatkan produktivitas. Aplikasi Octo Smart dibuat khusus untuk *Relationship Manager*, *Sales*, *Branch Manager*, dan *Senior Branch Manager*, dan tenaga penjualan di cabang CIMB Niaga terpilih, yang

menawarkan cara tercepat dan optimal untuk memberikan arahan, penerimaan prospek, dan pemantauan prospek.



Gambar 3.4. Tampilan *Home* Pada Aplikasi Octo Smart

3.2.4. Aktifitas *Automation Testing* Terhadap *Feature* Octo Smart (Minggu 7-21)

Pemberian tugas mandiri mulai diberikan sejak minggu ke 7. Tugas mandiri ini diberikan oleh mentor berupa *automation testing* pada *feature*

aplikasi Octo Smart yang nantinya ditentukan oleh mentor. Kegiatan *automation testing* sudah termasuk pembuatan skenario, pembuatan *test case*, hingga eksekusi serta pembuatan laporan dari hasil *testing.* Adapun *feature* yang dilakukan *automation testing* beserta hasilnya adalah sebagai berikut.

Tabel 3.2. *Feature* Yang Dikerjakan Selama Kegiatan Magang

No.	Nama *Feature*	Skenario	Status
1	*Change Security Code*	*Successfully change security code*	PASS
		Fail to change security code because input doesn't satisfy requirements	
		Fail to change security code because user enter wrong current security code	
2	*Add Activity on a Lead*	*Successfully add activity on a lead*	PASS
		Fail to add activity because status is either	

No.	Nama *Feature*	Skenario	Status
		already final, not in consecutive order, or back to previous status	
3	*Sort Leads*	*Successfully sort leads list based on aging and leads name alphabetically (descending – ascending)*	PASS
4	*Filter Leads*	*Successfully filter leads list according to lead's source, product, or status*	PASS
5	*Lead's Portfolio*	*Successfully create portfolio on a lead for individual or non-individual customer*	PASS
		Successfully link an existing portfolio to a lead for individual or non-individual customer	

No.	Nama *Feature*	Skenario	Status
		Fail to create a new portfolio because the alternative ID is already used for individual or non-individual customer	
6	*Call Report*	*Successfully create a call report on a lead for NTB or ETB customer*	PASS
7	*Onboard SME Lending*	*Successfully submit loan application form for individual or non-individual customer*	PASS
8	*ERWL*	*Successfully create a ERWL list and submit report creation to business manager*	PASS
9	*RAC SME Retail*	*Successfully check a customer eligibility for a loan*	PASS
10	*Simulator*	*Successfully simulate a loan to a customer to*	PASS

No.	Nama *Feature*	Skenario	Status
		check whether the customer's eligible or not eligible for a loan	
		Successfully link a simulator to leads	
11	*Product Recommendation*	*Successfully get a list of product recommendation for customer based on a survey*	PASS
		Successfully link a product recommendation result to leads	
		Successfully share an e-brochure after getting product recommendation result	
		Successfully edit customer answer on the related survey	

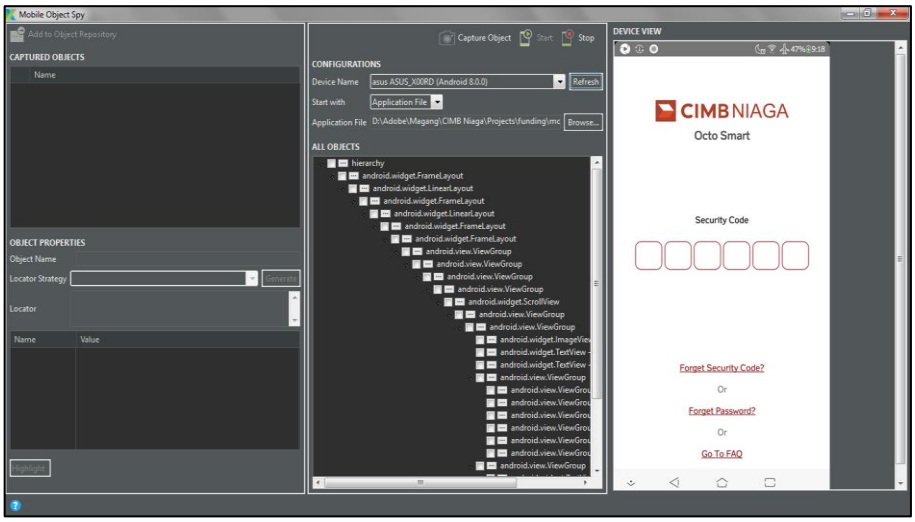
No.	Nama *Feature*	Skenario	Status
12	*Update Leads*	*Successfully update a lead then assign the lead to another user*	PASS
13	*Manage Team & Leads*	*Successfully update one of the team's lead*	PASS
		Successfully see leads list managed under a specific director's division	
14	*Share Product Info & Campaign*	*Successfully share a product information to a customer*	PASS
		Successfully share a campaign information to a customer	
15	*Review ERWL*	*Successfully review and give approval to an ERWL account*	PASS
		Successfully review and give revision to an ERWL account	

No.	Nama *Feature*	Skenario	Status
16	*Review Call Report*	*Successfully review and give approval to a call report*	PASS
		Successfully review and give revision to a call report	
17	*Onboard Non-Individu*	*Successfully register NTB customer to a microsite account for data registration*	PASS
		Successfully get notified if a customer is ETB based on NPWP checking	
18	*Handle Revision in ERWL*	*Successfully revise an ERWL record after being reviewed by manager*	PASS
19	*Handle Revision in Call Report*	*Successfully revise a call report after being reviewed by manager*	PASS

No.	Nama *Feature*	Skenario	Status
20	*Incentive*	*Successfully lookup incentive amount as a direct sales user*	PASS
21	*Handle Revision in Onboard Funding*	*Successfully revise onboard funding submission after being reviewed by manager for ETB, ETB BO, NTB, NTB BO customer*	
22	*Review Onboard Lending*	-	-
23	*Review Onboard Funding*	*Successfully review and give approval to an onboard funding submission*	PASS
		Successfully review and give revision to an onboard funding submission	
		Successfully cancel the onboard funding submission	

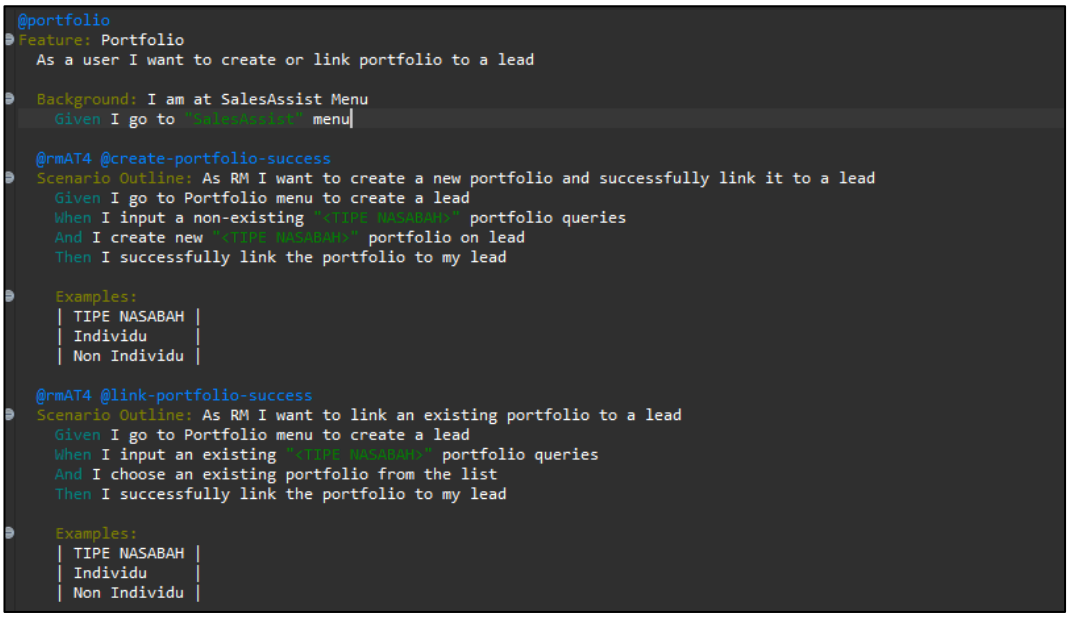
No.	Nama *Feature*	Skenario	Status
24	*Handle Revision in Onboard Consumer*	*Successfully revise onboard consumer submission after being reviewed by manager*	PASS
25	*Watchlist / EWA*	-	-
26	*Review Watchlist / EWA*	-	-

Tahapan pertama dalam melakukan *automation testing* adalah mengambil *object-object* pada halaman aplikasi. Setiap *feature* mungkin akan melewati halaman dan form yang berbeda namun ada juga yang sama. Untuk memastikan agar tidak ada *object* yang duplikat, maka *object* di *capture* per halaman yang ada di dalam aplikasi saja menggunakan Katalon Mobile Object Spy.



Gambar 3.5. Katalon Mobile Object Spy

Setelah *object* didapatkan, langkah selanjutnya adalah menulis skrip *gherkin* pada file *feature*. Dengan menggunakan *framework* Cucumber, istilah *test case* diubah ke dalam *steps* atau tahapan-tahapan yang membentuk suatu skenario uji. *Steps* ditulis dalam bahasa Inggris yang formal agar mudah dimengerti oleh para *stakeholder* non-teknis. Berikut contoh file *feature* Portfolio.



```
@portfolio
Feature: Portfolio
  As a user I want to create or link portfolio to a lead

  Background: I am at SalesAssist Menu
    Given I go to "SalesAssist" menu

  @rmAT4 @create-portfolio-success
  Scenario Outline: As RM I want to create a new portfolio and successfully link it to a lead
    Given I go to Portfolio menu to create a lead
    When I input a non-existing "<TIPE NASABAH>" portfolio queries
    And I create new "<TIPE NASABAH>" portfolio on lead
    Then I successfully link the portfolio to my lead

    Examples:
    | TIPE NASABAH |
    | Individu     |
    | Non Individu |

  @rmAT4 @link-portfolio-success
  Scenario Outline: As RM I want to link an existing portfolio to a lead
    Given I go to Portfolio menu to create a lead
    When I input an existing "<TIPE NASABAH>" portfolio queries
    And I choose an existing portfolio from the list
    Then I successfully link the portfolio to my lead

    Examples:
    | TIPE NASABAH |
    | Individu     |
    | Non Individu |
```

Gambar 3.6. Penulisan Skenario Dalam Bahasa Gherkin Untuk *Feature* Portfolio

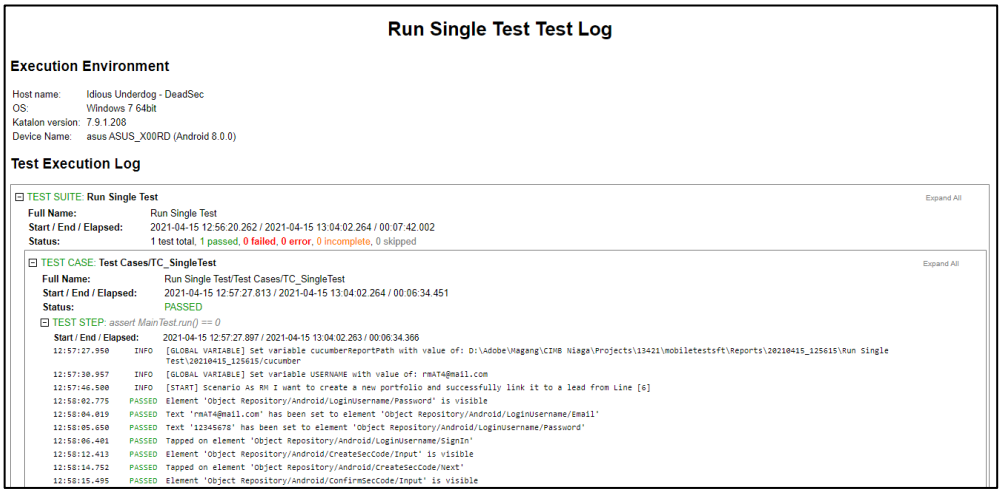
Jika skrip *gherkin* sudah siap, setiap *step* akan berperan sebagai *test case* yang harus dipenuhi agar keseluruhan skenario dapat berjalan dengan baik. Kemudian, file *feature* tersebut akan ditautkan dengan sebuah file lain yang berisi skrip *groovy* atau disebut dengan *step definition.* File ini berisikan kumpulan *step-step* yang ada di file *feature* terkait dan digunakan untuk

menjalankan aktifitas *automation.* Berikut contoh *step definition* untuk *feature* Portfolio.

```
package stepdefs

import cucumber.api.java.en.And
import cucumber.api.java.en.Then
import cucumber.api.java.en.When
import pages.LeadDetail
import pages.LeadsDataFormPage
import pages.SalesAssist
import pages.portfolio.CreateLeads
import pages.portfolio.CreateNewPortfolio
import pages.portfolio.PilihLeads
import pages.portfolio.PortfolioQuery
import pages.portfolio.PortfolioResult


public class Portfolio {
    @And("I go to Portfolio menu to create a lead")
    def I_go_to_Portfolio_menu_to_create_a_lead(){
        SalesAssist.clickPortfolio()
        PilihLeads.chooseNewLeads()
        //PilihLeads.chooseExistLeads()
        PilihLeads.clickNext()

        CreateLeads.InputFullName("test porto")
        CreateLeads.InputPhone("08123213242")
        CreateLeads.ChooseCategoryProduct("SME Funding")
        CreateLeads.ChooseProductOfInterest("Tabungan Usaha Individual")
        CreateLeads.InputEstimateAmounts("800000")
        CreateLeads.ChoosenthETB("Not Sure")
        CreateLeads.ClickButtonCREATELEADS()

    }

    @When("I input a non-existing {string} portfolio queries")
    def I_input_a_non_existing_portfolio_queries(String tipeNasabah){
        PortfolioQuery.isDisplayed()
```

Gambar 3.7. Penulisan Step Definition Dalam Bahasa Groovy Untuk *Feature* Portfolio

Ketika semua *keywords*, *step definition*, dan file *feature* sudah siap, langkah selanjutnya adalah melakukan eksekusi *automation testing.* Hasil dari *execution* ini akan dimuat ke dalam sebuah laporan dalam bentuk HTML. Skrip pembuatan laporan sudah disediakan oleh mentor sehingga tidak perlu membuatnya lagi. Berikut adalah contoh hasil eksekusi *testing* pada *feature* Portfolio.



Run Single Test Test Log

Execution Environment

Host name: Idious Underdog - DeadSec
OS: Windows 7 64bit
Katalon version: 7.9.1.208
Device Name: asus ASUS_X00RD (Android 8.0.0)

Test Execution Log

TEST SUITE: **Run Single Test** — Expand All

Full Name: Run Single Test
Start / End / Elapsed: 2021-04-15 12:56:20.262 / 2021-04-15 13:04:02.264 / 00:07:42.002
Status: 1 test total, 1 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

TEST CASE: **Test Cases/TC_SingleTest** — Expand All

Full Name: Run Single Test/Test Cases/TC_SingleTest
Start / End / Elapsed: 2021-04-15 12:57:27.813 / 2021-04-15 13:04:02.264 / 00:06:34.451
Status: PASSED

TEST STEP: *assert MainTest.run() == 0*

Start / End / Elapsed: 2021-04-15 12:57:27.897 / 2021-04-15 13:04:02.263 / 00:06:34.366

```
12:57:27.950    INFO    [GLOBAL VARIABLE] Set variable cucumberReportPath with value of: D:\Adobe\Magang\CIMB Niaga\Projects\13421\mobiletestsft\Reports\20210415_125615\Run Single Test\20210415_125615/cucumber
12:57:30.957    INFO    [GLOBAL VARIABLE] Set variable USERNAME with value of: rmAT4@mail.com
12:57:46.500    INFO    [START] Scenario As RM I want to create a new portfolio and successfully link it to a lead from Line [6]
12:58:02.775    PASSED  Element 'Object Repository/Android/LoginUsername/Password' is visible
12:58:04.019    PASSED  Text 'rmAT4@mail.com' has been set to element 'Object Repository/Android/LoginUsername/Email'
12:58:05.650    PASSED  Text '12345678' has been set to element 'Object Repository/Android/LoginUsername/Password'
12:58:06.481    PASSED  Tapped on element 'Object Repository/Android/LoginUsername/SignIn'
12:58:12.413    PASSED  Element 'Object Repository/Android/CreateSecCode/Input' is visible
12:58:14.752    PASSED  Tapped on element 'Object Repository/Android/createseccode/Next'
12:58:15.495    PASSED  Element 'Object Repository/Android/ConfirmSecCode/Input' is visible
```

Gambar 3.8. *Report* Hasil Eksekusi *Automation Testing* Pada *Feature* Portfolio

Apabila semua *testing* berhasil, tidak ada *bug* dari sisi skrip *testing* ataupun dari sisi aplikasi yang diuji maka pekerjaan dapat di *push* dan *merge* (setelah diulas oleh mentor) ke dalam repositori Gitlab yang disediakan untuk QA khusus Octo Smart. Siklus *testing* akan berjalan terus dengan tahapan yang sama pada *feature-feature* yang lainnya.

3.2.5. Melakukan *Adjustment Fix* (Minggu 22-23)

Setelah menyelesaikan tugas mandiri yang diberikan mentor berupa pembuatan *automation testing* untuk *feature* yang ada di dalam Octo Smart, tim *developer* Octo Smart mengeluarkan aplikasi versi terbaru dimana terdapat beberapa perbedaan di dalamnya. Maka dari itu, mentor menugaskan untuk melakukan penyesuaian terhadap *automation testing* terhadap beberapa *feature* yang sudah dibuat sebelumnya. Beberapa *feature* yang disesuaikan antara lain.

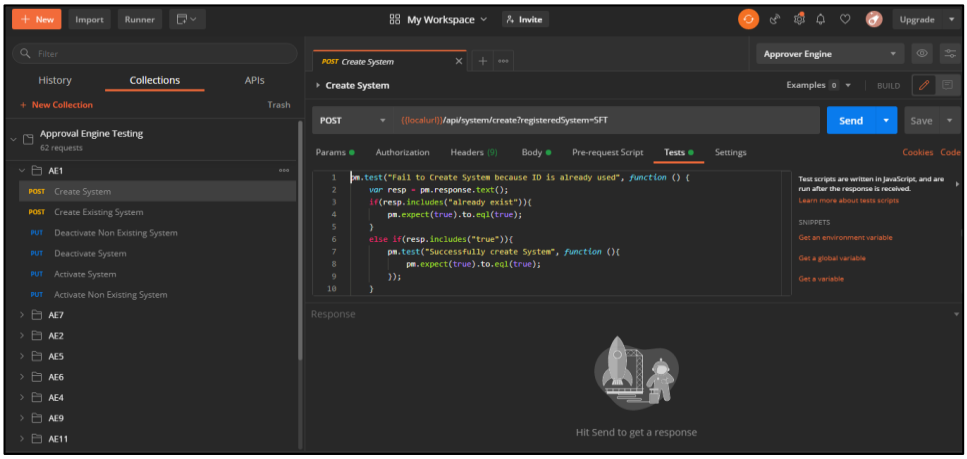
Tabel 3.3. Penyesuaian *Feature* Pada APK Baru

No.	Nama *Feature*	Penyesuaian
1	*Onboard Funding*	Perubahan *test data*
2	*Onboard Consumer*	Perubahan *test data*
		Penambahan *method* dan *test case* untuk skenario pada produk CC
		Penambahan *method* dan *test case* untuk skenario pada produk PL
3	*Portfolio*	Perubahan *test data*
		Penambahan skenario untuk nasabah *non-individual*

3.2.6. Belajar Melakukan API *Testing*, Memulai *Sprint* Baru (Minggu 24-26)

Dalam periode sebulan terakhir sebelum magang berakhir, mentor dari tim *intern developer* meminta bantuan untuk melakukan *API testing* terhadap projek yang akan dikerjakan oleh mereka yaitu membuat sebuah *approval engine.* Setelah *user story* ditentukan pada *sprint planning* baru, *developer* mulai membangun API. Sebagai *tester*, dari *user story* yang sudah ada, akan disiapkan skenario uji yang ditulis dalam bahasa JavaScript menggunakan Postman. *Developer* akan menyediakan *end-point* untuk diuji dan *tester* akan menulis skrip *test case* untuk menguji kesesuaian input-

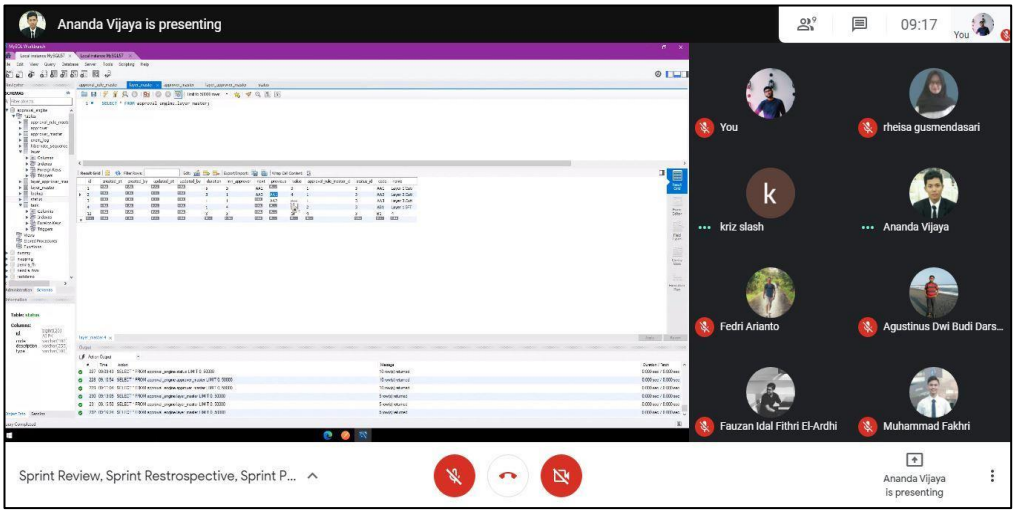
output terhadap kriteria yang telah ditentukan dalam *sprint planning*. Berikut adalah contoh skrip *testing* yang terdapat di Postman.



Gambar 3.9. Skrip *Testing* Salah Satu *End-Point* Pada *User Story* AE1

3.2.7. Demo *Testing* (Minggu 27)

Pada minggu terakhir magang, tepatnya di akhir minggu para mentor mengadakan *sprint review* yang diikuti semua tim *intern* dan beberapa perwakilan tim *internal* CIMB Niaga. *Sprint review* dipimpin langsung oleh *scrum master* dari tim *intern*. Para *developer* akan mendemokan API yang telah mereka bangun setelah itu baru diikuti oleh *tester* yang akan mendemokan hasil *testing* per *user story* yang telah ditentukan.



Gambar 3.10. Sprint Review Serta Demo Hasil Sprint yang Dilaksanakan Melalui Google Meet

3.3. Kendala yang Dihadapi

Selama menjalankan kegiatan kerja magang di PT. Bank CIMB Niaga, Tbk., terdapat beberapa kendala yang ditemukan. Beberapa diantaranya adalah:

a. Adanya pandemi COVID-19 mengharuskan seluruh karyawan bekerja *work from home* sehingga mahasiswa mengalami kesulitan dalam hal mencari informasi dan berkomunikasi langsung ke rekan kerja lain.

b. Untuk mengakses Jira membutuhkan VPN (*Virtual Private Network*) yang hanya dapat diakses oleh karyawan tetap. Jira merupakan sebuah perangkat lunak yang digunakan untuk mengelola pekerjaan khususnya suatu projek [5]. Sehingga mahasiswa harus bergantung kepada mentor untuk melakukan *automation testing* tanpa mengetahui *background* aplikasi, *user story* yang dikerjakan, *feature* apa yang membutuhkan atau siap di *testing*, akses *test data*, bahkan *version control* aplikasi Octo Smart.

c. Penggunaan Katalon Studio serta *framework* Cucumber BDD merupakan sesuatu yang baru bagi mahasiswa, sehingga harus dipelajari sendiri.

3.4. Solusi atas Kendala

Terkait dengan mengatasi beberapa kendala yang dihadapi selama melakukan kegiatan kerja magang di PT. Bank CIMB Niaga, Tbk. Berikut adalah solusi yang dilakukan:

a. Selama kegiatan kerja magang, mahasiswa melakukan *session* pribadi menggunakan Google Meet kepada mentor apabila terdapat hal yang sulit untuk dimengerti atau melakukan diskusi mengenai alur dari *feature* yang akan di *testing*.
b. Mengikuti perkembangan *automation testing* dengan *squad teste*r lain agar tidak tertinggal *update* setiap minggu dan juga mendiskusikan kendala dan *progress* setiap hari ke *scrum master* dan tim *intern* yang lain.
c. Meminta referensi yang paling relevan untuk mempelajari Katalon Studio dan *framework* Cucumber kepada mentor yang kemudian dapat dengan mudah dipahami dan dipelajari sendiri.